CLAIMS

What is claimed is:

1. A method comprising:

permitting a thread to be interrupted during a first critical region;

interrupting the thread with an interrupt routine; and

selectively setting the thread to restart at a beginning of the first critical region in response to an indication that the thread is working in a critical region.

2. The method of Claim 1, wherein the setting the thread to start at the beginning of the first critical region comprises:

setting a stack pointer and program counter so that the interrupted thread restarts at the beginning of the first critical region.

3. The method of Claim 1, wherein the first critical region includes the acts of:

selectively incrementing a count of a number of times the first critical region has been interrupted in response to the first critical region having been previously interrupted; and

selectively saving program stack conditions of the interrupted thread in response to the thread not having been interrupted during the first critical region more than a maximum permitted number of times.

4. The method of Claim 3, wherein the first critical region further includes the acts of:

setting a flag to indicate that a critical region is entered;

performing critical region interrupt recovery work in response to an indication that the first critical region was previously interrupted; and

attempting to complete first critical region work.

5. The method of Claim 4, wherein the critical region interrupt recovery work corrects errors that arise from previously interrupting the first critical region prior to its completion.

6. The method of Claim 4, wherein the first critical region further includes the acts of:

selectively setting a flag to indicate that critical region work is not being performed in response to the critical region work completing; and

setting to zero the count of the number of times the first critical region was interrupted.

7. The method of Claim 1 further comprising:

marking the thread as having been interrupted during the first critical region.

8. The method of Claim 1, further comprising indicating that the thread is not operating in a critical region.

9. A method comprising:

permitting a thread to be interrupted during a first critical region;

selectively incrementing a count of a number of times the first critical region has been interrupted in response to the first critical region having been previously interrupted; and

selectively saving program stack conditions of the interrupted thread in response to the thread not having been interrupted during the first critical region more than a maximum permitted number of times.

10. The method of Claim 9, further comprising:

setting a flag to indicate that a critical region is entered;

selectively performing critical region interrupt recovery work in response to an indication that the first critical region was previously interrupted; and

clearing a flag indicating the first critical region was previously interrupted.

11. The method of Claim 10, wherein the critical region interrupt recovery work corrects errors that arise from previously interrupting the first critical region prior to its completion.

12. The method of Claim 10, further comprising:

commencing work in the first critical region.

13. The method of Claim 12, further comprising:

selectively setting a flag to indicate that critical region work is not being performed in response to the critical region work completing; and

resetting a counter of times the first critical region was interrupted to zero.

14. The method of Claim 9, further comprising:

setting a flag to indicate that a critical region is entered; and

selectively commencing work in the first critical region in response to an indication that the first critical region was not previously interrupted.

15. The method of Claim 14, further comprising:

selectively setting a flag to indicate that critical region work is not being performed in response to the critical region work completing; and

resetting a counter of times the first critical region was interrupted to zero.

16. A method comprising:

selectively disabling interrupts of a critical region in response to the critical region having been interrupted more than a permitted number of times; and

commencing critical region work.

17. The method of Claim 16, further comprising:

selectively setting a flag to indicate that critical region work is not being performed in response to the critical region work completing;

selectively resetting a counter of times the critical region was interrupted to zero in response to the critical region work completing; and

re-enabling interrupts of the critical region.

18. A method comprising:

permitting a critical region to be interrupted; and

selectively saving stack conditions of an interrupted critical region in response to the critical region not having been previously interrupted.

19. The method of Claim 18, further comprising:

commencing work in the critical region.

20. The method of Claim 19, further comprising:

selectively setting a flag to indicate that critical region work is not being performed in response to the critical region work completing; and

selectively resetting a counter of times the critical region was interrupted to zero in response to the critical region work completing.

21. An article comprising a storage medium, the storage medium comprising machine readable instructions stored thereon to:

permit a thread to be interrupted during a first critical region;

interrupt the thread with an interrupt routine; and

selectively set the thread to restart at a beginning of the first critical region in response to an indication that the thread operates in a critical region.

22. The article of Claim 21, wherein the instructions to set the thread to restart at the beginning of the first critical region comprises instructions to:

set a stack pointer and program counter so that the interrupted thread restarts at the beginning of the first critical region.

23. The article of Claim 21, wherein the first critical region includes instructions to:

selectively increment a count of a number of times the first critical region has been interrupted in response to the first critical region having been previously interrupted; and

selectively save program stack conditions of the interrupted thread in response to the thread not having been interrupted during the first critical region more than a maximum permitted number of times.

24. The article of Claim 23, wherein the first critical region further includes instructions to:

set a flag to indicate that a critical region is entered;

perform critical region interrupt recovery work in response to an indication that the first critical region was previously interrupted; and

attempt to complete first critical region work.

25. The article of Claim 24, wherein the critical region interrupt recovery work includes instructions to correct errors that arise from previously interrupting the first critical region prior to its completion.

26. The article of Claim 24, wherein the first critical region further includes instructions to:

selectively set a flag to indicate that critical region work is not being performed in response to the critical region work completing; and

set to zero the count of the number of times the first critical region was interrupted.

27. The article of Claim 21 further comprising instructions to mark the thread as having been interrupted during the first critical region.

28. The article of Claim 21, further comprising instructions to indicate that the thread is not operating in a critical region.

29. An apparatus comprising:

an I/O controller device comprising a processor and a memory device, the I/O controller device comprising logic to:

permit a thread to be interrupted during a first critical region;

interrupt the thread with an interrupt routine; and

selectively set the thread to restart at a beginning of the first critical region in response to an indication that the thread operates in a critical region.

30. The apparatus of Claim 29, wherein the logic to set the thread to restart at the beginning of the first critical region comprises logic to:

set a stack pointer and program counter so that the interrupted thread restarts at the beginning of the first critical region.

31. The apparatus of Claim 29, wherein the first critical region includes logic to:

selectively increment a count of a number of times the first critical region has been interrupted in response to the first critical region having been previously interrupted; and

selectively save program stack conditions of the interrupted thread in response to the thread not having been interrupted during the first critical region more than a maximum permitted number of times.

32. The apparatus of Claim 31, wherein the first critical region further includes logic to:

set a flag to indicate that a critical region is entered;

perform critical region interrupt recovery work in response to an indication that the first critical region was previously interrupted; and

attempt to complete first critical region work.

33. The apparatus of Claim 32, wherein the critical region interrupt recovery work includes logic to correct errors that arise from previously interrupting the first critical region prior to its completion.

34. The apparatus of Claim 32, wherein the first critical region further includes logic to:

selectively set a flag to indicate that critical region work is not being performed in response to the critical region work completing; and

set to zero the count of the number of times the first critical region was interrupted.

35. The apparatus of Claim 29 further comprising logic to mark the thread as having been interrupted during the first critical region.

36. The apparatus of Claim 29, further comprising logic to indicate that the thread is not operating in a critical region.

37. A system comprising:

a first storage subsystem and a second storage subsystem;

a first circuit card including first circuitry capable of being coupled to the first storage subsystem, wherein the first circuitry further includes an I/O controller device, the I/O controller device comprising:

a processor;

a memory device, wherein the memory device includes code segments that instruct the processor to:

permit a thread to be interrupted during a first critical region,

interrupt the thread with an interrupt routine, and

selectively set the thread to restart at a beginning of the first critical region in response to an indication that the thread operates in a critical region; and

a second circuit card including second circuitry capable of being coupled to the second storage subsystem.

38. The system of Claim 37, wherein the second circuit includes an I/O controller device.

39. The system of Claim 37, wherein the first storage subsystem and the second storage subsystem each comprise one or more respective mass storage devices.

40. The system of claim 37, wherein:

the first storage subsystem comprises a redundant array of inexpensive disks (RAID); and

the second storage subsystem comprises a tape mass storage system.

41. The system of claim 37, further comprising:

a circuit board coupled to the bus, wherein the circuit board comprising a bus, memory, and a host processor; and

the first circuit card and the second circuit card are capable of being coupled to the bus.

42. The system of Claim 41, wherein the bus complies with PCI.